IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Lanzhong Wang	Confirmation No.: 6921
Serial No.: 10/829,488	Group Art Unit: 2167
Filed: April 22, 2004	Examiner: Lovel, Kimberly M.
For: SYSTEM AND METHODS INVOLVING A DATA STRUCTURE SEARCHABLE WITH O(logN) PERFORMANCE	Atty. Docket: 100203738-1

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In response to the Final Office Action mailed on October 16, 2007, the applicant appeals as follows:

This brief contains items under the following headings as required by 37 CFR §41.37 and MPEP §1206:

I. Real Party In Interest
II. Related Appeals, Interferences and Judicial Proceedings
III. Status of Claims
IV. Status of Amendments
V. Summary of Claimed Subject Matter
VI. Grounds of Rejection to be Reviewed on Appeal
VII. Argument
VIII. Claims
IX. Evidence
X. Related Proceedings

Appendix A	Claims
Appendix B	Evidence
Appendix C	Related Proceedings

(I) REAL PARTY IN INTEREST

The real party in interest in the above-referenced patent application is the Hewlett-Packard Development Company, L.P., 20555 SH 249 Houston, Texas 77070.

(II) RELATED APPEALS, INTERFERENCES AND JUDICIAL PROCEEDINGS

There are no related appeals, interferences or judicial proceedings currently known to the Appellants, Appellants' legal representatives or the assignee, which will directly affect, or be directly affected by, or have a bearing on, the Board's decision.

(III) STATUS OF CLAIMS

Claims 1-25 are pending and are rejected. The rejections of all claims are appealed.

(IV) STATUS OF AMENDMENTS

No amendments were filed or entered subsequent to the final rejection mailed October 16, 2007.

(V) SUMMARY OF THE CLAIMED SUBJECT MATTER

The invention as claimed is summarized below with reference numerals and references to the specification and drawings. The invention is broadly set forth in the language corresponding to independent claims 1, 6, 17, and 21. Discussions about elements of the invention can be found at least in the locations in the specification and drawings cited in the claims below.

1. A data structure that is stored on a computer-readable medium comprising:

a sorted portion (24, 110) that contains a plurality of entries that are sorted into an order; [paragraphs 15, 16, 23; figs. 1, 2, 5]

an unsorted portion (25, 120) that contains a plurality of entries (23) that have not been sorted; [paragraphs 16, 23; figs 1, 2, 5] and

a boundary (21, 130) that separates the sorted portion (24, 110) and the unsorted portion (25, 120); [paragraphs 16, 23; figs. 1, 2, 5]

wherein the sorted portion (24, 110) of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion. [paragraphs 16, 17, 21, 23, 25, 26; figs. 3, 4]

6. A method of using a container (20, 100) that comprises a sorted portion (24, 110) that contains a plurality of entries that are sorted into an order, an unsorted portion (120) that contains a plurality of entries (25) that have not been sorted, and a boundary (21, 130) that separates the sorted portion (24, 110) and the unsorted portion (120), the method comprising: [paragraphs 16, 23; figs. 1, 5]

receiving a search request (31, 401) that comprises a requested value; [paragraph 21, 23, 26; figs. 3, 4]

searching (32, 402) the sorted portion (24, 110) of the container (20, 100) for the requested value with O(logN) performance; [paragraphs 15, 17, 21, 23, 26; figs. 2, 3, 4]

adding an entry (414) to the unsorted portion (25, 120) during the searching; [paragraphs 17, 21, 23; fig. 4] and

returning a stored value (34) of the container (100) if there is a match of the stored value and the requested value. [paragraphs 21, 25, 26; figs. 3, 4]

17. A computer program product having a computer-readable medium (804, 806) having computer program logic recorded thereon for inserting a new value into a container (20, 100) that comprises a sorted portion (24, 110) that contains a plurality of entries that are sorted into an order, an unsorted portion (25, 120) that contains a plurality of entries (23) that have not been sorted, and a boundary (21, 130) that separates the sorted portion (24, 110) and the unsorted portion (25, 120), the computer program product comprising: [paragraph 16; figs. 1, 5]

code for searching the sorted portion (24, 32, 110) of the container (20, 100) for the new value with O(logN) performance (402); [paragraphs 15, 16, 17, 23, 26; figs. 2, 3, 4]

code for searching the unsorted portion (110) of the container (100) if no match is found in the search of the sorted portion (110) with O(N) performance (409); [paragraphs 21, 23; fig. 4] and

code for inserting the new value into the container (100) if no match is found in the search of the unsorted portion (414). [paragraph 21, 25; figs. 4]

21. A computer system (800) for managing data objects, comprising:

memory means (804, 806) for storing said data objects; [paragraphs 15, 16; figs. 1, 5, 8]

means for identifying a boundary (130) within said memory means (804, 806) for storing, wherein data objects stored in a first portion (110) of said memory means (804, 806) defined by said boundary (130) are stored in an ordered manner and data objects stored in a second portion (120) of said memory means (804, 806) defined by said boundary (130) are stored in an unordered manner; [paragraphs 15, 16, and 23; figs. 1, 4, and 8]

means for searching said first portion (110) according to O(logN) performance to locate an identified object. [paragraphs 15, 17, 21; figs. 4]

(VI) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-5 and 17-20 were rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter.

Claims 1-4, 6-15, 17-19, and 21-25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Steinman (U.S. 5,850,538) in view of Rajasekaran (U.S. 2005/0256890).

Claims 5, 16, and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Steinman (U.S. 5,850,538) in view of Rajasekaran (U.S. 2005/0256890), and further in view of Watkins (U.S. 6,901,207).

The appellant contends that the rejections are in error.

(VII) ARGUMENT

I. Rejection of Claims 1-5 and 17-20 under 35 U.S.C. §101

The office action states that the claimed invention is directed toward non-statutory subject matter. In summary, the office action further states that the claims are directed toward forms of energy, which the office action states are not patentable. The applicants disagree with the claims being defined as forms of energy. Claim 1 is reprinted as follows for convenience:

> A data structure that is stored on a computer-readable medium comprising:
>
> a sorted portion that contains a plurality of entries that are sorted into an order;
>
> an unsorted portion that contains a plurality of entries that have not been sorted; and

a boundary that separates the sorted portion and the unsorted portion; wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion.

Claim 1 is directed toward a data structure stored on a computer-readable medium. Thus, claim 1 is not directed toward a form of energy as set forth in the final office action. Rather, claim 1, is directed toward a computer-readable medium with a data structure stored thereon.

The Response to Arguments section of the final office action states that claim 1 is directed to a wave, signal, or form of energy. The office action then applied the holding in *In re Nuijten*, 84 USPQ2d 1495 (CAFC 2007) to hold that claim 1 is not proper subject matter per 35 U.S.C. §101. The applicants note that the subject matter in *Nuijten* is solely related to a waveform having watermarks associated therewith. Claim 1, on the other hand, includes a tangible computer-readable medium. Thus, claim 1 is not a signal-type claim as set forth in *Nuijten*. In fact, as stated below and in Footnote 6 of *Nuijten*, the Court in *Nuijten* has not dealt with a storage medium and the PTO allowed the claims in *Nuijten* that contained storage mediums.

The applicants note that *Nuijten* defines *machine* at 1501. More specifically, the Court cited *Burr v. Duryee*, 68 U.S. (1 Wall.) 531, 570 (1863) as follows:

> The Supreme Court has defined the term 'machine' as 'a concrete thing, consisting of parts, or of certain devices and combination of devices.'
>
> *Id.* at 1501

Based on the holdings, claim 1 is at least directed toward a machine in that a computer-readable medium is a concrete thing consisting of parts. Accordingly, claim 1 is directed toward statutory subject matter per *Nuijten*.

In addition to defining machine, the Court in *Nuijten* defines *manufacture* at 1502. The Court held that the signal alone in *Nuijten* was not a manufacture per 35

U.S.C. §101. However, in footnote 6, the Court stated the following in reference to the signal being devoid of any semblance of permanence during transmission:

> Of course, such a signal could be stored for later use, but the result of such storage would be a "storage medium" containing the signal. Such a storage medium would likely be covered by allowed Claim 15 of Nuijten's application, which is not before us on appeal.

Thus, the USPTO has allowed Nuijten's claim that included a storage medium. In addition, the Court has not directed its decision toward a storage medium. However, relying on the holding of the USPTO in this matter, a storage medium, such as a computer-readable medium must be proper subject matter as it is allowable per *Nuijten*.

Claim 17 is independent and, like claim 1, includes a computer-readable medium, which is allowable subject matter as set forth above. Claims 4-5 and 18-20 are dependent on claims 1 and 17 and are deemed to be directed toward proper subject matter.

Based on the foregoing, the applicant contends that the rejection of claims are improper and requests reversal of the rejections.

II. Rejection of Claims 1-4, 6-15, 17-19, and 21-25 Under 35 U.S.C. §103(a)

Claims 1-4, 6-15, 17-19, and 21-25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Steinman (U.S. 5,850,538) in view of Rajesekaran (U.S. 2005/0256890).

Claim 1

Claim 1 is independent and is reprinted as follows for convenience:

> A data structure that is stored on a computer-readable medium comprising:
>
> a sorted portion that contains a plurality of entries that are sorted into an order;
>
> an unsorted portion that contains a plurality of entries that have not been sorted; and
>
> a boundary that separates the sorted portion and the unsorted portion;
>
> wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion.

Claim 1 includes a data structure comprising two portions, a sorted portion and an unsorted portion. In summary, the applicants contend that Steinman does not disclose a single data structure comprising two portions as claimed. According to the office action, these portions are disclosed by Steinman at column 6, line 65 to column 7, line 9. This section of Steinman discloses an event horizon and two lists as follows:

> In order to exploit the improved event horizon for event list management algorithms in accordance with the present invention, it is assumed that as new events are generated, **they are not immediately sent back into the**

main priority queue data structure, but instead are collected in an unsorted temporary holding queue. The event with the earliest time tag in this temporary queue is tracked. When the next event to be processed is in the temporary queue (i.e., the event horizon has been crossed), the queue is sorted (a binary merge sort algorithm is easily performed on linked lists) and then merged back into the main priority queue data structure. (Emphasis added)

Based on this section of Steinman, a "main priority queue data structure" stores sorted data and an "unsorted temporary holding queue" stores unsorted data. Thus, sorted data is stored in one data structure and unsorted data is stored in the temporary holding queue, which is different than the main priority queue. Therefore, Steinman does not disclose a data structure comprising the sorted portion and the unsorted portion as claimed in claim 1. Rather, there are two different data structures. There is no mention of these elements in Rajesekaran. Accordingly, the references, taken individually or in combination, fail to disclose all the elements of claim 1 (namely two data structures) and, therefore, cannot render claim 1 obvious.

The office action refers to the event horizon of Steinman as being the boundary of claim 1. Again, the applicants disagree with this holding. The event horizon taught by Steinman is a time stamp and is not a boundary as claimed in claim 1. Reference is made to the above-cited section of Steinman wherein the event with the earliest time tag in the temporary queue is tracked. When the next event to be processed is in the temporary queue, the queue is sorted, meaning that the horizon has been crossed. Accordingly, the event horizon is a timestamp and not a boundary as claimed. The event horizon is similarly described at column 6, lines 14-24, as a timestamp rather than a boundary.

Based on the foregoing, Steinman does not disclose the boundary of claim 1. There is no mention of the boundary in Rajesekaran. Therefore, the references, taken

individually or on combination, do not disclose all the elements of claim 1 and cannot render claim 1 obvious.

Claim 1 also recites "wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion." According to the office action, Steinman discloses adding an entry to the unsorted portion, but there is no mention of adding the entry while searching the sorted portion as recited in claim 1. The cited portions of Steinman disclose searching a list and adding an item to a list as separate functions, not together. Rajesekaran is provided to teach O(LogN) performance. There is no mention of Rajesekaran disclosing "wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion" as claimed. Therefore, neither reference, taking individually or in combination, teach this element of claim 1.

Based on the foregoing, neither reference, taken individually or in combination, teach all the elements or claim 1 or render clam 1 obvious. Accordingly, the applicants request reversal of the rejection.

CLAIMS 2-4

Claims 2-4 stand or fall with claim 1 solely for the purposes of this appeal.

CLAIM 6

Claim 6 is independent and states the following:

> A method of using a container that comprises a sorted portion that contains a plurality of entries that are sorted into an order, an unsorted portion that contains a plurality of entries that have not been sorted, **and a boundary that separates the sorted portion and the unsorted portion**, the method comprising:

receiving a search request that comprises a requested value;

searching the sorted portion of the container for the requested value with O(logN) performance;

adding an entry to the unsorted portion during the searching; and

returning a stored value of the container if there is a match of the stored value and the requested value.

Some portions of claim 6 that are not disclosed by either Steinman, Rajasekaran, or their combination have been printed above in bold text.

As stated above, neither Steinman nor Rajasekaran disclose the boundary as claimed. Rather, Steinman discloses an event horizon that functions as a time stamp and does not boundaries as claimed. Accordingly, neither reference, nor their combination disclose or suggest the claimed boundary.

Furthermore and as stated above, neither reference discloses adding an entry to the unsorted portion of a container during a search of the sorted portion.

Based on the foregoing, neither reference, taken individually or in combination teaches or suggests the elements of claim 6. Accordingly, the references cannot render claim 6 obvious. Therefore, the applicants request reversal of the rejection.

CLAIMS 7-15

Claims 7-15 stand or fall with claim 6 solely for the purposes of this appeal.

CLAIM 17

Claim 17 is independent and is reprinted as follows for convenience:

A computer program product having a computer-readable medium having computer program logic recorded thereon for inserting a new value

into a container that comprises a sorted portion that contains a plurality of entries that are sorted into an order, an unsorted portion that contains a plurality of entries that have not been sorted, and **a boundary that separates the sorted portion and the unsorted portion**, the computer program product comprising:

code for searching the sorted portion of the container for the new value with O(logN) performance;

code for searching the unsorted portion of the container if no match is found in the search of the sorted portion with O(N) performance; and

code for inserting the new value into the container if no match is found in the search of the unsorted portion.

Some portions of claim 17 that are not disclosed by either Steinman, Rajasekaran, or their combination have been printed above in bold text.

As stated above, neither Steinman nor Rajasekaran disclose the boundary as claimed. Rather, Steinman discloses an event horizon that functions as a time stamp and does not mention boundary portions of a container as claimed. Accordingly, neither reference, nor their combination disclose or suggest the claimed boundary.

Based on the foregoing, neither reference, taken individually or in combination teaches or suggests the elements of claim 17. Accordingly, the references cannot render claim 17 obvious. Therefore, the applicants request reversal of the rejection.

CLAIMS 18 AND 19

Claims 18 and 19 stand or fall with claim 17 solely for the purposes of this appeal.

CLAIM 21

Claim 21 is independent and is reprinted as follows for convenience:

A computer system for managing data objects, comprising:
memory means for storing said data objects;
means for identifying a boundary within said memory means for storing, wherein data objects stored in a first portion of said memory means defined by said boundary are stored in an ordered manner and data objects stored in a second portion of said memory means defined by said boundary are stored in an unordered manner; and
means for searching said first portion according to O(logN) performance to locate an identified object.

Some portions of claim 21 that are not disclosed by either Steinman, Rajasekaran, or their combination have been printed above in bold text.

As stated above, neither Steinman nor Rajasekaran disclose the boundary. Therefore, the references cannot disclose means for identifying a boundary as claimed. Rather, Steinman discloses an event horizon that functions as a time stamp and does not teach boundaries as claimed. Accordingly, neither reference, nor their combination disclose or suggest the claimed means for identifying a boundary.

Based on the foregoing, neither reference, taken individually or in combination teaches or suggests the elements of claim 21. Accordingly, the references cannot render claim 21 obvious. Therefore, the applicants request reversal of the rejection.

CLAIMS 22-25

Claims 22-25 stand or fall with claim 21 solely for the purposes of this appeal.

III. Rejection of Claims 5, 16, and 20 Under 35 U.S.C. §103(a)

Claims 5, 16, and 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Steinman (U.S. 5,850,538) in view of Rajesekaran (U.S. 2005/0256890), and further in view of Watkins (U.S. 6,901,207).

Claims 5, 16, and 20 are dependent and will stand or fall with their respective independent claims.

In view of the above, all of the pending claims are now believed to be in condition for allowance and a notice to that effect is earnestly solicited.

Respectfully submitted,
KLAAS, LAW, O'MEARA & MALKIN, P.C.

By: /Robert Nelson/
Robert Nelson
Registration No. 37,898
1999 Broadway, Suite 2225
Denver, CO 80202
(303) 298-9888

APPENDIX A - CLAIMS

What is claimed is:

1. (Previously presented) A data structure that is stored on a computer-readable medium comprising:

a sorted portion that contains a plurality of entries that are sorted into an order;

an unsorted portion that contains a plurality of entries that have not been sorted; and

a boundary that separates the sorted portion and the unsorted portion;

wherein the sorted portion of the data structure is searchable with O(logN) performance while an entry is added to the unsorted portion.

2. (Previously presented) The data structure of claim 1, wherein the sorted portion is searchable with a binary search.

3. (Previously presented) The data structure of claim 1, wherein the unsorted portion may be searched with an incremental search.

4. (Original) The data structure of claim 1, wherein the data structure may be sorted to form a new sorted portion that comprises the plurality of entries of the sorted portion and the plurality of entries of the unsorted portion, and the plurality of entries of the new sorted portion are sorted into an order.

5. (Original) The data structure of claim 1, wherein the data structure is associated with an occurrence model used in designing a circuit.

6. (Original) A method of using a container that comprises a sorted portion that contains a plurality of entries that are sorted into an order, an unsorted portion that

contains a plurality of entries that have not been sorted, and a boundary that separates the sorted portion and the unsorted portion, the method comprising:

receiving a search request that comprises a requested value;

searching the sorted portion of the container for the requested value with O(logN) performance;

adding an entry to the unsorted portion during the searching; and

returning a stored value of the container if there is a match of the stored value and the requested value.

7. (Original) The method of claim 6, wherein when there is not a match, the method further comprises:

returning a null value that indicates that there is no match with the requested value.

8. (Original) The method of claim 6, wherein when there is not a match, the method further comprises:

adding an entry to the unsorted portion corresponding to the search request.

9. (Original) The method of claim 6, wherein when there is not a match, the method further comprises:

determining whether unsorted items in the container exceed a predetermined threshold;

performing a sort operation on the container, if the predetermined threshold is exceeded, thereby forming a new sorted portion that comprises the plurality of entries of the sorted portion and the plurality of entries of the unsorted portion, and the plurality of entries of the new sorted portion are sorted into an order.

10. (Original) The method of claim 9, further comprises:

searching the new sorted portion of the container for the requested value; and

returning a stored value of the container if there is a match of the stored value and the requested value.

11. (Original) The method of claim 10, wherein searching the new sorted portion comprises:

searching with O(logN) performance.

12. (Original) The method of claim 6, wherein when there is not a match, the method further comprises:

searching the unsorted portion of the container for the requested value; and

returning a stored value of the container if there is a match of the stored value and the requested value.

13. (Original) The method of claim 12, wherein the unsorted portion may be searched with an incremental search.

14. (Original) The method of claim 6, wherein when there is not a match, the method further comprises:

determining whether a size of the unsorted portion is zero;

adding an entry to the unsorted portion corresponding to the search request if the unsorted portion is not zero.

15. (Original) The method of claim 14, wherein the size of the unsorted portion is zero, the method further comprises:

determining whether the requested value of the search request is greater than the value of the last entry of the sorted portion;

adding an entry to the unsorted portion corresponding to the search request if the requested value of the search request is not greater than the value of the last entry of the sorted portion;

adding an entry to the sorted portion corresponding to the search request if the requested value of the search request is greater than the value of the last entry of the sorted portion.

16. (Original) The method of claim 6, further comprises:

using the container in an occurrence model in designing a circuit.

17. (Original) A computer program product having a computer-readable medium having computer program logic recorded thereon for inserting a new value into a container that comprises a sorted portion that contains a plurality of entries that are sorted into an order, an unsorted portion that contains a plurality of entries that have not been sorted, and a boundary that separates the sorted portion and the unsorted portion, the computer program product comprising:

code for searching the sorted portion of the container for the new value with O(logN) performance;

code for searching the unsorted portion of the container if no match is found in the search of the sorted portion with O(N) performance; and

code for inserting the new value into the container if no match is found in the search of the unsorted portion.

18. (Original) The computer program product of claim 17, wherein the code for inserting comprises:

code for determining whether to insert the new value in the sorted portion or the unsorted portion of the container.

19. (Original) The computer program product of claim 17, further comprises:

code for sorting the unsorted portion and merging the sorted portion and the sorted unsorted portion into a new sorted portion, wherein the code for sorting is operative when the unsorted portion exceeds a predetermined criteria; and

code for searching the new sorted portion of the container for the new value with O(logN) performance.

20. (Original) The computer program product of claim 17, further comprises:

code for a circuit design.

21. (Previously presented) A computer system for managing data objects, comprising:

memory means for storing said data objects;

means for identifying a boundary within said memory means for storing, wherein data objects stored in a first portion of said memory means defined by said boundary are stored in an ordered manner and data objects stored in a second portion of said memory means defined by said boundary are stored in an unordered manner; and

means for searching said first portion according to O(logN) performance to locate an identified object.

22. The computer system of claim 21 further comprising:

means for searching said second portion for said identified object according to O(N) performance.

23. The computer system of claim 21 further comprising:

means for adding said identified object to said second portion when said means for searching said first portion and said means for searching said second portion do not locate said identified object.

24. The computer system of claim 21 further comprising:

means for merging data objects in said second portion into said first portion in an ordered manner; and

means for resetting said boundary in response to said means for merging.

25. The computer system of claim 24 wherein said means for merging is operable when a number of data objects in said second portion reaches a predetermined amount.

APPENDIX B - EVIDENCE

There is no evidence to be presented

APPENDIX C - RELATED PROCEEDINGS

There are no related proceedings.